MongoDB : Modeling – Administration – Java API

A. You will create a new database in the field of education (you will call the database forma). It will include information on teachers (with name, first name and mobile number) and information on courses (name, duration (in days)). In term of queries, we aim to retrieve information on the teacher giving a course. It is assumed that each teacher can give many courses and that there are many write operations on the database (write / read ratio on the base is close to 1), that is to say that the teachers of a courses can be modified frequently.
Based on this approach, model the database using 2 collections : course and teacher.
1. Propose an approach to rettrieve info on the the teacher of a course.
2. Record 2 documents for the teacher collection and 2 for the course collection. Relate them such that a teacher teaches at least on course.
3. Write the document enabling the retrieval of the teacher's last and first name given a course (without writing any constant value other than the course name).
4. Backup the database.
5. Look at the metadata of the dump
6. Remove a teacher and remove the 2 courses. Compared to a relational database, what do you observe ?
7. Restore the complete database with the - - drop option. Verify that you have 2 documents per colection. Do you see anything special ?

Additional information: MongoDB offers to create links between two documents (as we had to do between teachers and courses): DBRef. Please note, not all the functions of the foreign keys of a relational database are supported. In particular, the consistency of the database is not guaranteed when a linked document is deleted. It can be used in a program (most languages supported by MongoDB support DBRef). Read the following page to get more information :
https://docs.mongodb.com/manual/reference/database-references/

B. We will import a CSV dataset into a new database.
8. Import the movie information from the movies.csv file into the moviedb database. The first line corresponds to the fields of the file. The information will be stored in the movies collection.

9. Create an admin user with read / write rights on the admin database and the ability to administer all databases. Restart the server with the authentication option, then create a user with read-only rights on the movies database. Relaunch the console and log in with this new user. Read and write.

10. What is happening if we restart mongod without the - - auth option ?

C. Using the movie database, we will write a Java program that will manipulate the data. Create a Maven project and insert the following dependance in the pom.xml file:

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongo-java-driver</artifactId>
  <version>numero de version</version>
</dependency>
```

With the help of the Mongodb javadoc (https://mongodb.github.io/mongo-java-driver/3.6/javadoc/), create a first Java class that
11. creates a connection to the MongoClient on localhost and port 27017.
12. connects to the `moviedb` database
13. retrieves the first document of the movies collection and displays it in JSON.

14. retrieves the number of documents from the movies collection
15. Displays the list of movies released in 1995 using a mongoDB cursor (instance of MongoCursor).
16. Insert the movie Logan from your Java program
17. Update the Logan movies with the following release data : 03-03-2017.
18. Remove the Logan movie